第三讲

矩阵特征值计算

—— 应用：Google 网页排名

PageRank

- 网站排名是网络搜索引擎的核心
- PageRank 是著名网络搜索引擎 Google 用于评测一个网页“重要性”或“影响力”的一种方法。通过该方法，Google 将各个网站进行排名。用户进行相关搜索时，Google 会将符合条件的网站按排名顺序输出。
- PageRank 得分越大表示网页越重要。
- PageRank 算法中使用的数学知识包括：正矩阵性质、特征值和特征向量、幂迭代算法、Gauss-Seidel迭代算法等

本讲主要介绍 PageRank 算法的基本思想与模型，以及如何使用该算法对网站进行排名

有向图

有向图介绍

- **有向图的定义、相关术语和部分性质**

有向图是指由有限个元素的非空集合和它的不同元素构成的有序数对组成的结构。

图的基本元素：顶点（节点）和边（线、弧、枝）

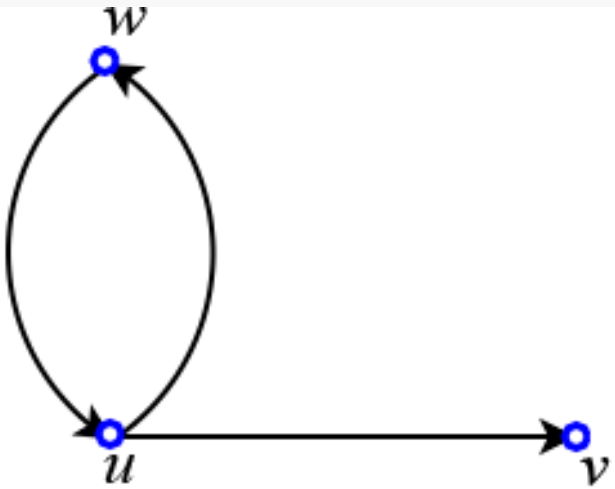
例：右图为一个有向图，记为 D，

其顶点组成的集合记为

V(D) = { u, v, w}

边组成的集合记为

A(D) = {(u,w), (w,u), (u,v)}



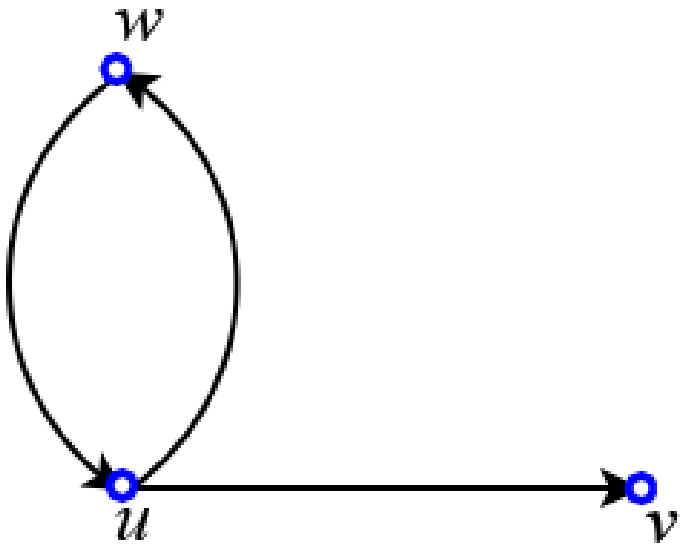
- 注：(u,w) 和 (w,u) 表示不同的边。

有向图相关术语

- 有向图 D 的**顶点集**的基数称为 **D 的阶**，记作 **p(D)**
- 边组成的集合的基数称为 **D 的大小**，记作 **q(D)**
- 顶点 **v** 的**出度** (out-degree) 是指从 **v** 邻接的顶点的个数，或以 **v** 为起点的边的条数，记作 **od(v)**
- 顶点 **v** 的**入度**(in-degree)是指 D 中邻接到 **v** 的顶点的个数，或以 **v** 为终点的边的条数，记作 **id(v)**

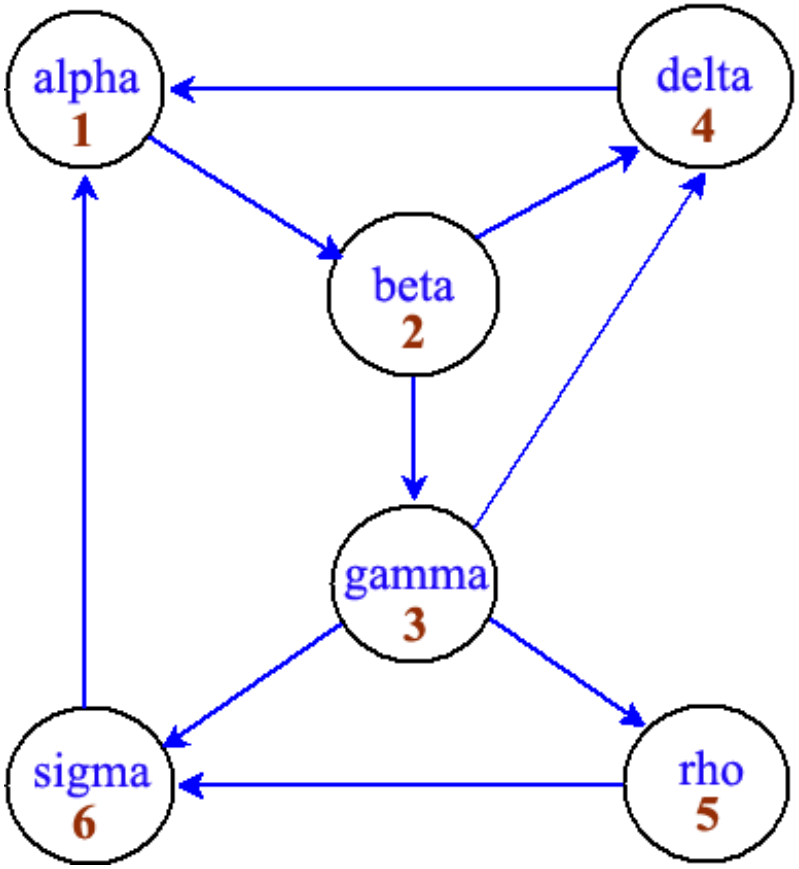
有向图举例

例：右下图为一个有向图，记为 D，则



- D 的阶: p(D)=3
- D 的大小: q(D)=3
- 顶点 u 的出度: od(u)=2
- 顶点 u 的入度: id(u)=1
- 顶点 v 的出度: od(v)=0
- 顶点 v 的入度: id(v)=1

有向图举例



例：左图中

p(D)=6，q(D)=9

序号	顶点	入度	出度
1	alpha	2	1
2	beta	1	2
3	gamma	1	3
4	delta	2	1
5	rho	1	1
6	sigma	2	1

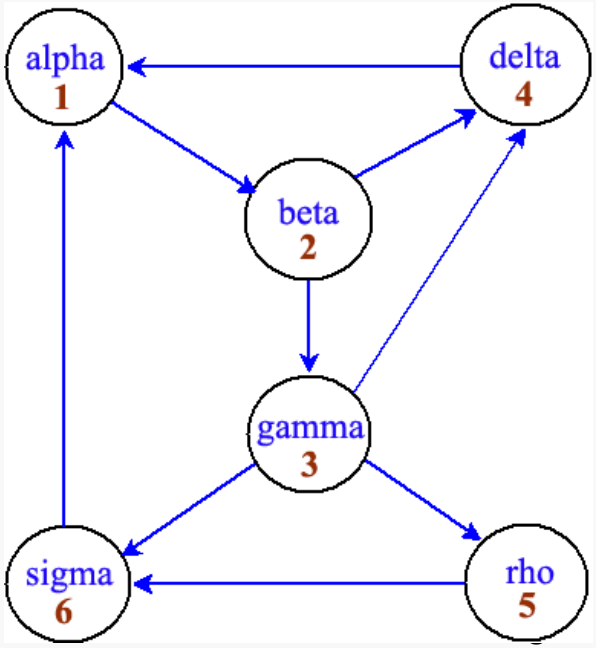
邻接矩阵

- 为研究需要，我们定义邻接矩阵

$$G = (g_{ij}), \quad \text{其中 } g_{ij} = \begin{cases} 1, & \text{如果存在从 } j \text{ 到 } i \text{ 的弧} \\ 0, & otherwise \end{cases}$$

例：对于右边的有向图，其邻接矩阵为

$$G = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$



邻接矩阵的性质

性质一： 定义行和 $r_i = \sum_j g_{ij}$ 和列和 $n_j = \sum_i g_{ij}$，则第 i 行的行和 r_i 就是第 i 个顶点的入度，第 j 列的列和 n_j 就是第 j 个顶点的出度。

行和 $\leftrightarrow$ 入度，列和 $\leftrightarrow$ 出度

性质二： $\sum_i r_i = \sum_j n_j = q(D)$ =边的个数

PageRank 数学模型

PageRank 的决定因素

Google 的 PageRank 是基于这样一个理论：

若 B 网页上有连接到 A 网站的链接 (称 B 为 A 的导入链接)，说明 B 认为 A 有链接价值，是一个“重要”的网站，此时 A网站可从 B网站分得一定的级别 (重要性)。

同时 A 的级别将平均分配给 A网站上的所有导出链接。

导入链接：链接到你网站的站点，即“外部链接”；

导出链接：网站上指向另外一个站点的链接。

在 PageRank 模型中，一个网站的级别（重要性）大致由下面两个因素决定：导入链接的数量和导入链接的级别（重要性）。

哪个网页最重要

如果我们将下面的有向图中的每个顶点看成一个网站，并把每条边看成是网站间的“超链接”，则此有向图就代表一个小型的网络，其中有 6 个网站和 9 个超链接。

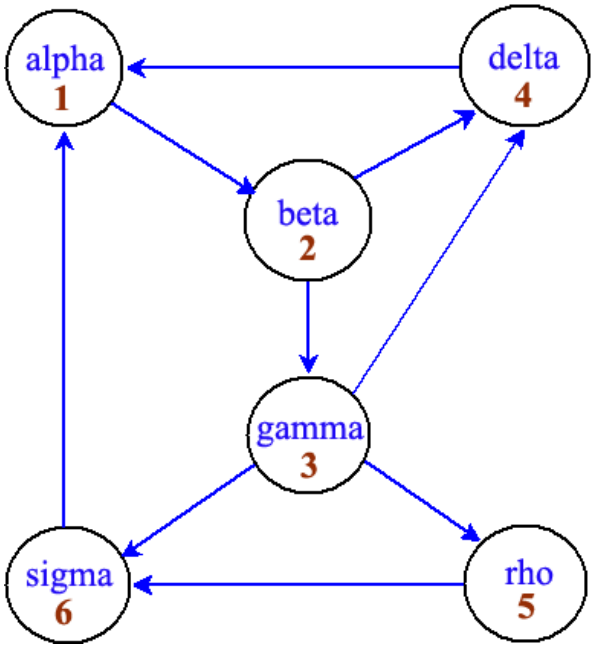
例： 这 6 个网站中哪个最重要？

看谁的导入链接多？

不太合理

重要性的决定因素：

- 导入链接的数量
- 导入链接的重要性



简化的 PageRank 算法

设 u 是某个网页，其级别（重要性）为 $r(u)$，记 F_u 为 u 的导出链接的集合，B_u 为 u 的导入链接的集合，$n_u = |F_u|$ 即是 u 的导出链接总数。

设 v 是 u 的一个导入链接，根据 PageRank 理论，u 从 v 处分得的级别（重要性）为 $r(v)/n_v$。将 u 从所有导入链接处分得的重要性相加，即为网页 u 的最终级别

$$r(u) = \sum_{v \in B_u} \frac{r(v)}{n_v}$$

简化的PageRank模型

设共有 m 个网页，分别编号为 1、2、3、...、m，它们的级别（重要性）分别记为 r_1、r_2、r_3、...、r_m，G 表示由这些网页组成的有向图的邻接矩阵。根据有向图理论：

$$r(u)=\sum_{v\in B_u}\frac{r(v)}{n_v} \Rightarrow r_i=\sum_{j=1}^{m}\frac{g_{ij}}{n_j}r_j$$

n_j：G 中第 j 列的列和

矩阵形式

$$r=G_m\cdot r$$

其中 $\begin{cases} r=(r_1,r_2,\ldots,r_m)^T \\ G_m=\{g_{ij}/n_j\} \end{cases}$

简化PageRank的问题

$$r = G_m \cdot r$$

- 易知 r 是 G_m 的对应于特征值为 1 的特征向量

矩阵 G_m 一定有特征值 1 吗？即上面的方程是否有解？

- 如果 $G = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$，则 $r_1 = r_2$ ，此时就无法进行排名

因此，我们需要对简化的 PageRank 进行改进！

改进的 PageRank

- 基本思想：首先给每个网页设置一个基本级别

设 $\eta(u)$ 是网页 u 的所获得的基本级别，则

$$x(u) = p\sum_{v\in B_u}\frac{x(v)}{n_v} + \eta(u)$$

其中：

- $x(u)$ 表示网页 u 的最终级别
- p 是一个加权系数，通常取 0.85 左右
- $\eta(u) = (1 - p)/m \equiv \delta$

改进的 PageRank

与前面的讨论相类似，将所有网页进行编号：

1、2、...、m

于是可以把右式改写为：

$$x(u) = p\sum_{v \in B_u} \frac{x(v)}{n_v} + \eta(u)$$

$$x_i = p\sum_{j=1}^{n} \frac{g_{ij}}{n_j} x_j + \delta \qquad (i = 1, 2, \ldots, m)$$

矩阵 形式

$$x = p \cdot G_m \cdot x + \delta \cdot e$$

$$\begin{pmatrix} x = (x_1, x_2, \ldots, x_n)^T \\ e = (1, 1, \ldots, 1)^T \end{pmatrix}$$

改进的 PageRank

$$x = p \cdot G_m \cdot x + \delta \cdot e$$

$$G_m = \{g_{ij} / n_j\} \Rightarrow G_m = G \cdot D$$

其中：$G = (g_{ij})$, $D = \mathrm{diag}\left(\frac{1}{n_1}, \frac{1}{n_2}, \ldots, \frac{1}{n_m}\right)$

$$x = p \cdot G \cdot D \cdot x + \delta \cdot e$$

改进的 PageRank

$$x = p \cdot G \cdot D \cdot x + \delta \cdot e$$

$$= p \cdot G \cdot D \cdot x + \delta \cdot e \cdot 1$$

$$= p \cdot G \cdot D \cdot x + \delta \cdot e \cdot e^T x$$

$$= \left(p \cdot G \cdot D + \delta \cdot e \cdot e^T \right) x$$

规定：$\sum_{i=1}^{n} x_i = 1$

$1 = \sum_{i=1}^{n} x_i = e^T x$

记 $A = p \cdot G \cdot D + \delta \cdot e e^T$

$$x = A \cdot x$$

x 是 A 的对应于特征值 $\lambda=1$ 的特征向量。

$$A=\begin{bmatrix} p\frac{g_{11}}{n_1}+\delta & p\frac{g_{12}}{n_2}+\delta & \cdots & p\frac{g_{1m}}{n_m}+\delta \\ p\frac{g_{21}}{n_1}+\delta & p\frac{g_{22}}{n_2}+\delta & \cdots & p\frac{g_{2m}}{n_m}+\delta \\ \vdots & \vdots & \ddots & \vdots \\ p\frac{g_{m1}}{n_1}+\delta & p\frac{g_{m2}}{n_2}+\delta & \cdots & p\frac{g_{mm}}{n_m}+\delta \end{bmatrix}_{m\times m}$$

- 矩阵 A 的两个重要性质：

  (1) $A>0$，即所有元素都是正数

  (2) A 的各列的列和等于 1

$$n_j=\sum_i g_{ij}$$

$$\delta=(1-p)/m$$

改进的 PageRank

$$A = p \cdot \begin{bmatrix} \frac{g_{11}}{n_1} & \frac{g_{12}}{n_2} & \cdots & \frac{g_{1m}}{n_m} \\ \frac{g_{21}}{n_1} & \frac{g_{22}}{n_2} & \cdots & \frac{g_{2m}}{n_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{g_{m1}}{n_1} & \frac{g_{m2}}{n_2} & \cdots & \frac{g_{mm}}{n_m} \end{bmatrix} + \delta$$

若矩阵 G 中存在 0 列，即存在 j 使得对所有的 i 有 $g_{ij} = 0$，则将导致 $n_j = 0$， 此时规定：

$$\begin{cases} g_{ij} = 1 \quad (i = 1, 2, \ldots, m) \\ n_j = m \end{cases}$$

改进的 PageRank

$$x = A \cdot x$$

x 满足：$\sum_{i=1}^{n} x_i = 1$

问：上述方程组的解是否存在？

答：上述方程组存在唯一的解！（且均为正数）

理由：Perron-Frobnius 定理（证明略）

A 的谱半径

问：$\lambda = 1$ 是 A 的特征值吗？

A 的各列的列和等于 1 $\Leftrightarrow$ $e^T \cdot A = e^T$

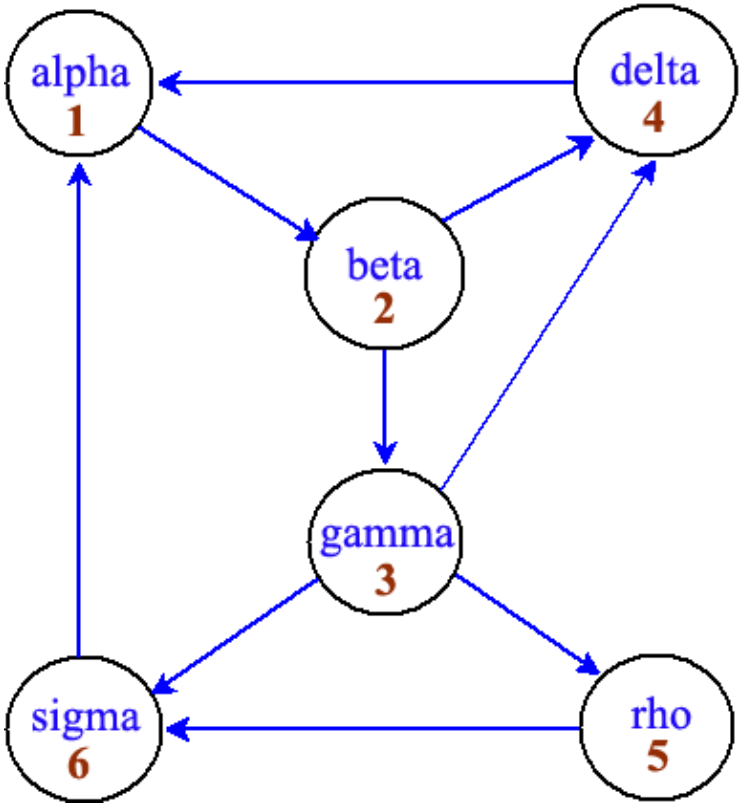
$\Rightarrow$ $e^T(I - A) = 0$

$\Rightarrow$ $|I - A| = 0$ $\Rightarrow$ $\lambda = 1$ 是 A 的特征值

事实上，我们有结论：$\lambda = 1$ 是 A 的惟一的模最大特征值，即其他特征值的模均严格小于 1。

网页排名举例

例：用改进的 PageRank 算法计算下面的小型网络中各网页的排名，其中取 $p=0.85$。



序号	顶点	入度	出度
1	alpha	2	1
2	beta	1	2
3	gamma	1	3
4	delta	2	1
5	rho	1	1
6	sigma	2	1

网页排名举例

```
clear;      % Eig11.m
p = 0.85;  % 此处 p 也可以取其它数值
G = [0 0 0 1 0 1; 1 0 0 0 0 0; 0 1 0 0 0 0; ...
     0 1 1 0 0 0; 0 0 1 0 0 0; 0 0 1 0 1 0];
n = size(G,1);
sn = sum(G);      % 提取每列的列和
D = diag(1./sn); % 生成对角矩阵
delta = (1-p)/n;
A = p*G*D + delta;
[v,d] = eig(A); % 计算 A 的特征值与特征向量
r = v(:,idx);   % 最大特征值所对应的特征向量
r = r./sum(r);  % 归一化
[x,index] = sort(r,'descend'); % 排序

% 输出结果
```

数値算法

幂法

当矩阵 A 的阶数很大时，无法直接计算其特征值和特征向量，此时需要使用迭代算法。

$x = Ax$，x 满足：$\sum_{i=1}^{n} x_i = 1$

- 幂法

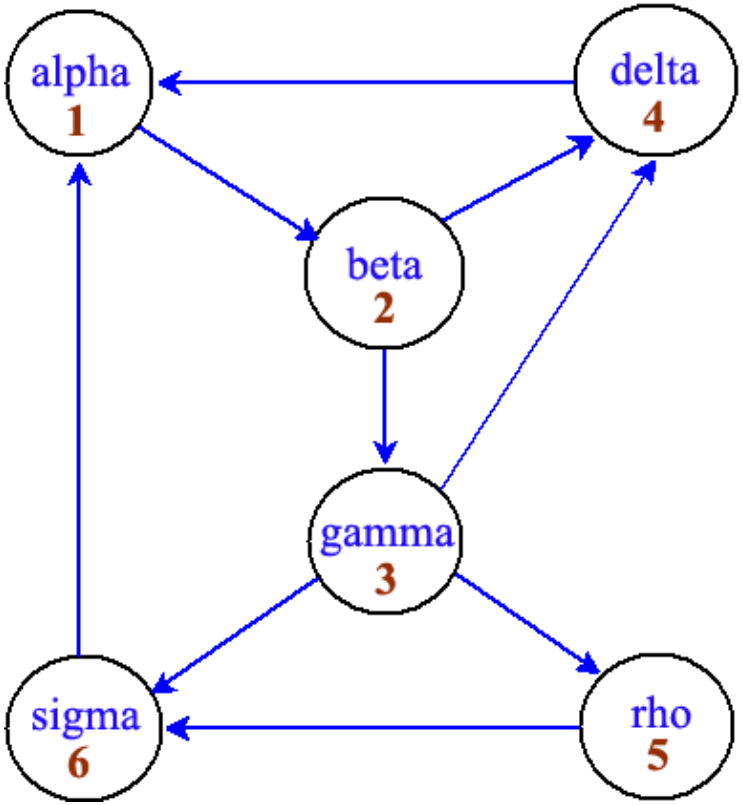
1) 输入矩阵 A 和初始向量 $v_0 > 0$，以及精度 tol

2) 计算：$u_{k+1} = Av_k$，$v_{k+1} = \dfrac{u_{k+1}}{\mathrm{sum}(u_{k+1})}$；

3) 如果 $|v_{k+1} - v_k| < \mathrm{tol}$，则令 $x = v_{k+1}$ 并停机，否则转第二步。

幂法举例

例：采用幂迭代法计算下面各网页的排名，其中 p=0.85。



幂法举例

```
clear; % Eig12.m
tol = 1e-4;  p = 0.85;
G = [ 0 0 0 1 0 1; 1 0 0 0 0 0; 0 1 0 0 0 0;
      0 1 1 0 0 0; 0 0 1 0 0 0; 0 0 1 0 1 0 ];
n = size(G,1);
sn = sum(G,1);    D = diag(1./sn);
delta = (1-p)/n;
A = p*G*D + delta;
x = ones(n,1)/n; % 迭代初始向量
z = zeros(n,1);
k = 0;  % 记录迭步数
while max(abs(x-z)) > tol    % 幂法
   z = x;
   x = A*x;
   k = k+1;
end
[x1,index]=sort(x,'descend');
```

一个问题

在前面给出的程序中，如果矩阵 G 中存在某一列的列和为零，怎么办？

此时规定：

$$\begin{cases} g_{ij} = 1 \quad (i = 1, 2, \ldots, m) \\ n_j = m \end{cases}$$

- 一个修改后的 Matlab 程序（`Eig13.m`）

另一个解决方案见 `Eig14.m`，它充分利用稀疏矩阵的性质，当矩阵规模较大时，能大大减少运算量。